Verification of MOESI Cache Coherence Protocol using UVM

Lakshmi Narayanan, Varadharajan (MS Electrical Engineering) Manoharan, Anushree (MS Electrical Engineering)

Introduction

“A hardware design is only as good as its verification”. Creating a verification model for design takes as much time as developing the design. With more number of past components being reused in the current designs nowadays, it is important to build reusable verification IPs. The Universal Verification Methodology is an industry-wide approach in creating reusable Verification IPs. A UVM testbench is created by following specified guidelines laid out by the methodology. The guidelines facilitate consistency between different verification groups, allowing the sharing of environments between multiple platforms. These guideline-recommended coding approaches allow users to debug with ease, leading to a shorter time to build the verification model.

This project is to develop a reusable verification IP for a MOESI cache coherence protocol. With the increase in the number of multiprocessor systems with a shared cache line, the need for a stable cache coherence protocol is also increasing. The remodeling of cache coherence protocols takes place periodically, and the new, upgraded protocols are extensions of previous protocols. Developing a new verification environment is difficult, time-consuming, and expensive. The project focuses on creating a reusable, upgradable Verification IP for the MOESI Cache coherence protocol.

Methodology

Verification of the MOESI Cache Coherence Protocol using UVM

Universal Verification Methodology, usually referred to as UVM, is a framework in System Verilog, that consists of a set of base classes. The base classes let the user create a modular, reusable verification environment. The UVM is a message-based Verification methodology, converting all the pin level translations into messages. The UVM consists of components built in a hierarchical form, each performing different operations. In this project, a verification hierarchy consisting of a stimulus generation unit and a checker unit was developed.

Developing the Stimulus Generation unit

The stimulus generation unit generates random sequences consisting of a sequence and a driver component classes, the sequence class generates sequences and sends them to the design under test through the driver. We generated sequences with each state of the MOESI cache protocol as initial states and used one-hot encoding on a variable (sel_input) to give one operation as input at all times to see if the design works according to the state transitions given in the specification. The five different scenarios created are as follows:

1. Scenario-1 *RESET and input state = I*
2. Scenario-2 *RESET and input state = I*
3. Scenario-3 *RESET and input state = E*
4. Scenario-4 *RESET and input state = O*
5. Scenario-5 *RESET and input state = M*

Randomize function was used to the variable sel_input and 50 random sequences are generated for each scenario with different inputs high each time with the one-hot encoder.

Scoreboard Unit

The scoreboard in this verification model consists of a reference model for the state machine of the design, a scoreboard for each of the states M, O, E, S, I, and scoreboards to process the intermediate states. The active monitor (input) and the passive monitor (output) send message objects to the respective scoreboards. The reference model scoreboard gets the sequence objects from the input monitor and predicts the outcomes based on the sequence.

The checker scoreboard accepts message objects from the output monitor from the passive agent and transmits the message to each of the state checking scoreboards. The result compilation and checking occur in the checker scoreboard of the respective states.

After the result checking, the UVM generates a report of all the errors and information about the verification. The result interpretation helps in improving the design to work according to the specification.

Analysis and Results

The picture shows the UVM report generated and shows that currently the design works without any error.

Summary/Conclusions

In this project, we proposed using UVM to develop a verification model for the MOESI cache coherence protocol. The verification model developed has the features to generate random test sequences to cover all cases in the MOESI protocol and scoreboards to verify if the results from the design match with the expected results from the reference model created. The results of the project can be seen through a waveform in the gtkwave application and also the UVM report generated. The verification model can be further modified in the future to handle additional states if a new cache coherence protocol is developed. Thus, a reusable Verification model was built using UVM to verify the MOESI cache coherence protocol.

Key References


Acknowledgements

We would also like to thank Prof Hui Wong for accepting to be our advisor and for mentoring us through summer, making us do useful work for his project. Also, we would like to thank Prof. Birsen Sirkeci for allowing us to do this project work for our masters.

We would like to express our deepest gratitude for Prof. Morris Jones for being our mentor and guiding us throughout our graduate study, almost in all of the semesters. His constant support, teaching and immense knowledge has helped us move forward towards our graduation. This project would never have been possible for us without him, he created a passion towards digital verification.